Fortify Audit Workbench

Developer Workbook

FortifyCodecNetv4

Table of Contents

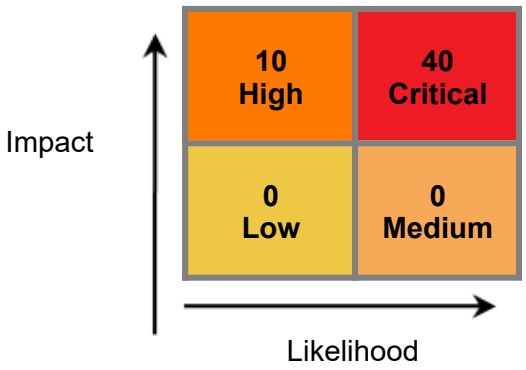
Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the FortifyCodecNetv4 project audit. The information contained in this workbook is targeted at project managers and developers.

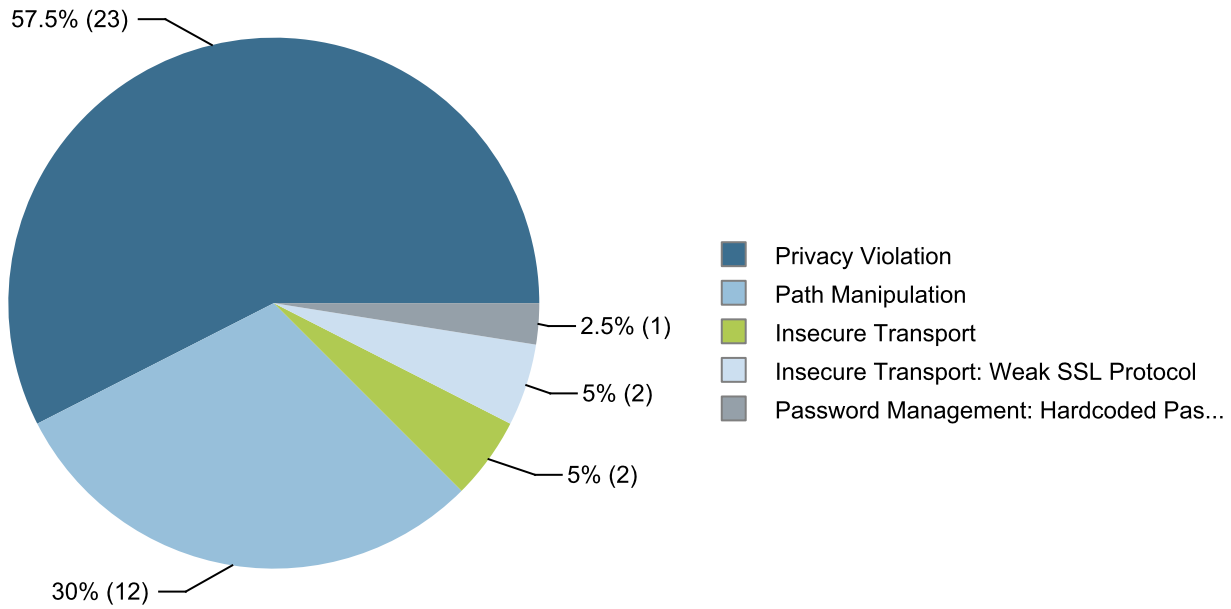
This section provides an overview of the issues uncovered during analysis.

Project Name:	FortifyCodecNetv4
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



Top Ten Critical Categories



Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Nov 13, 2025 12:32 PM	**Engine Version:**	25.3.0.0014
Host Name:	CodecFS	**Certification:**	VALID
Number of Files:	280	**Lines of Code:**	80,946

Rulepack Name	Rulepack Version
Fortify Secure Coding Rules, Community, Cloud	2025.3.0.0007
Fortify Secure Coding Rules, Community, Universal	2025.3.0.0007
Fortify Secure Coding Rules, Core, Cloud	2025.3.0.0007
Fortify Secure Coding Rules, Core, JavaScript	2025.3.0.0007
Fortify Secure Coding Rules, Core, Universal	2025.3.0.0007
Fortify Secure Coding Rules, Extended, Configuration	2025.3.0.0007
Fortify Secure Coding Rules, Extended, Content	2025.3.0.0007
Fortify Secure Coding Rules, Extended, JavaScript	2025.3.0.0007

Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Cookie Security: Overly Broad Path	0	2 / 2	0	0	2 / 2
Insecure Transport	2 / 2	0	0	0	2 / 2
Insecure Transport: Weak SSL Protocol	2 / 2	0	0	0	2 / 2
Password Management: Empty Password	0	6 / 6	0	0	6 / 6
Password Management: Hardcoded Password	1 / 1	2 / 2	0	0	3 / 3
Path Manipulation	12 / 12	0	0	0	12 / 12
Privacy Violation	23 / 23	0	0	0	23 / 23

Results Outline

Cookie Security: Overly Broad Path (2 issues)

Abstract

A cookie with an overly broad path can be accessed through other applications on the same domain.

Explanation

Developers often set cookies to be accessible from the root context path ("/"). This exposes the cookie to all web applications on the domain. Because cookies often carry sensitive information such as session identifiers, sharing cookies across applications can cause a vulnerability in one application to compromise another application. **Example 1:** Imagine you have a forum application deployed at `http://communitypages.example.com/MyForum` and the application sets a session ID cookie with the path "/" when users log in to the forum. For example:

```
cookie_options = {};
cookie_options.path = '/';
...
res.cookie('important_cookie', info, cookie_options);
```

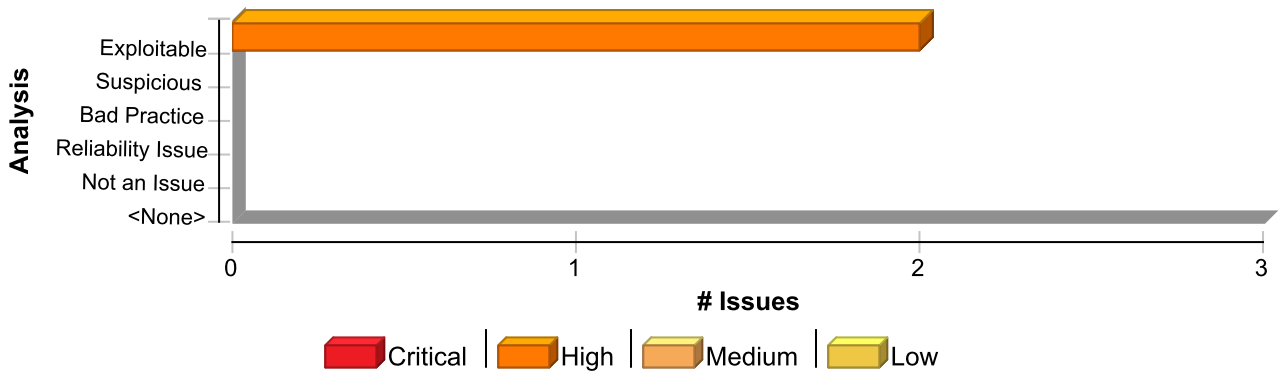
Suppose an attacker creates another application at `http://communitypages.example.com/EvilSite` and posts a link to this site on the forum. When a user of the forum clicks this link, the browser will send the cookie set by `/MyForum` to the application running at `/EvilSite`. By stealing the session ID, the attacker can compromise the account of any forum user that browsed to `/EvilSite`. In addition to reading a cookie, it might be possible for attackers to perform a Cookie Poisoning attack by using `/EvilSite` to create its own overly broad cookie that overwrites the cookie from `/MyForum`.

Recommendation

Make sure to set cookie paths to be as restrictive as possible. **Example 2:** The following code shows how to set the cookie path to "`/MyForum`", which fixes `Example 1`.

```
cookie_options = {};
cookie_options.path = '/MyForum';
...
res.cookie('important_cookie', info, cookie_options);
```

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Cookie Security: Overly Broad Path	2	0	0	2
Total	**2**	**0**	**0**	**2**

Cookie Security: Overly Broad Path — High

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/auth.controller.js, line 51 (Cookie Security: Overly Broad Path)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: Read path
From: verify2FA
File: Downloads/CodecNetv3/Backend/controllers/auth.controller.js:55

```
52  httpOnly: true, // Prevent JavaScript access (XSS protection)
53  secure: true, // Only transmit over HTTPS (was conditional)
54  sameSite: 'strict', // CSRF protection
55  path: '/', // Explicit path scope (accessible site-wide)
56  maxAge: 7 * 24 * 60 * 60 * 1000 // 7 days
57  });
58
```

Sink Details

Sink: ~JS_Generic.cookie()
Enclosing Method: verify2FA()
File: Downloads/CodecNetv3/Backend/controllers/auth.controller.js:51
Taint Flags: COOKIE_BROAD_PATH

```
48
49  // SECURITY FIX (PATCH 55): Secure cookie settings (CWE-1004, CWE-614)
50  // SECURITY FIX (PATCH 56): Explicit path attribute (CWE-284)
51  res.cookie('refreshToken', refresh_token, {
52  httpOnly: true, // Prevent JavaScript access (XSS protection)
53  secure: true, // Only transmit over HTTPS (was conditional)
54  sameSite: 'strict', // CSRF protection
```

Downloads/CodecNetv3/Backend/controllers/auth.controller.js, line 84 (Cookie Security: Overly Broad Path)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Cookie Security: Overly Broad Path — High

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/auth.controller.js, line 84 (Cookie Security: Overly Broad Path)

Audit Details

Analysis Exploitable

Source Details

Source: Read path
From: refreshToken
File: Downloads/CodecNetv3/Backend/controllers/auth.controller.js:88

```
85  httpOnly: true, // Prevent JavaScript access (XSS protection)
86  secure: true, // Only transmit over HTTPS (was conditional)
87  sameSite: 'strict', // CSRF protection
88  path: '/', // Explicit path scope (accessible site-wide)
89  maxAge: 7 * 24 * 60 * 60 * 1000 // 7 days
90  });
91
```

Sink Details

Sink: ~JS_Generic.cookie()
Enclosing Method: refreshToken()
File: Downloads/CodecNetv3/Backend/controllers/auth.controller.js:84
Taint Flags: COOKIE_BROAD_PATH

```
81
82  // SECURITY FIX (PATCH 55): Secure cookie settings (CWE-1004, CWE-614)
83  // SECURITY FIX (PATCH 56): Explicit path attribute (CWE-284)
84  res.cookie('refreshToken', result.refresh_token, {
85  httpOnly: true, // Prevent JavaScript access (XSS protection)
86  secure: true, // Only transmit over HTTPS (was conditional)
87  sameSite: 'strict', // CSRF protection
```

Insecure Transport (2 issues)

Abstract

The call uses an insecure protocol instead of a secure protocol to communicate with the server.

Explanation

All communication over HTTP, FTP, or gopher is unauthenticated and unencrypted. It is therefore subject to compromise, especially in the mobile environment where devices frequently connect to unsecured, public, wireless networks using WiFi connections. **Example 1:** The following example reads data using the HTTP protocol (instead of using HTTPS).

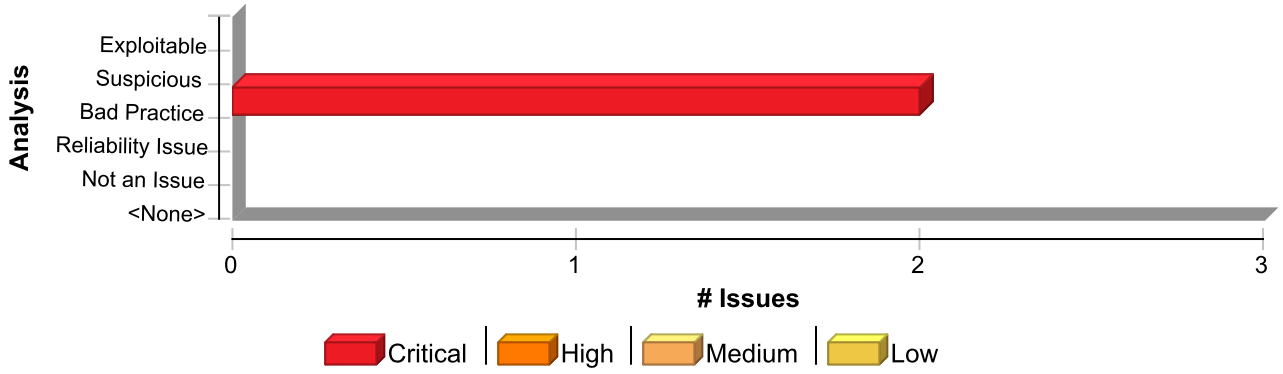
```
var http = require('http');
...
http.request(options, function(res){
...
});
...
```

The incoming `http.IncomingMessage` object,`res`, may have been compromised as it is delivered over an unencrypted and unauthenticated channel.

Recommendation

Use secure protocols such as HTTPS to exchange data with the server whenever possible.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Transport	2	0	0	2
Total	**2**	**0**	**0**	**2**

Insecure Transport — Critical

Package: Downloads.CodecNetv3.Backend

Downloads/CodecNetv3/Backend/index.js, line 101 (Insecure Transport)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Insecure Transport — Critical

Package: Downloads.CodecNetv3.Backend

Downloads/CodecNetv3/Backend/index.js, line 101 (Insecure Transport)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FunctionPointerCall: createServer
Enclosing Method: startHTTPServer()
File: Downloads/CodecNetv3/Backend/index.js:101
Taint Flags:

```
98
99 // Function to start HTTP server (development/fallback)
100 const startHTTPServer = () => {
101   const httpServer = http.createServer(app);
102
103   httpServer.listen(PORT, () => {
104   if (NODE_ENV === 'production') {
```

Downloads/CodecNetv3/Backend/server.js, line 189 (Insecure Transport)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FunctionPointerCall: createServer
Enclosing Method: startHTTPServer()
File: Downloads/CodecNetv3/Backend/server.js:189
Taint Flags:

```
186   console.error(' Please ensure MongoDB is running at:', process.env.MONGODB_URI);
187   }
188
189   const httpServer = http.createServer(app);
190
191   httpServer.listen(PORT, '0.0.0.0', () => {
192   if (NODE_ENV === 'production') {
```

Insecure Transport: Weak SSL Protocol (2 issues)

Abstract

The SSLv2, SSLv23, SSLv3, TLSv1.0, and TLSv1.1 protocols contain flaws that make them insecure and should not be used to transmit sensitive data.

Explanation

The Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols provide a protection mechanism to ensure the authenticity, confidentiality, and integrity of data transmitted between a client and web server. Both TLS and SSL have undergone revisions resulting in periodic version updates. Each new revision is designed to address the security weaknesses discovered in previous versions. Use of an insecure version of TLS/SSL weakens the data protection strength and might allow an attacker to compromise, steal, or modify sensitive information. Weak versions of TLS/SSL might exhibit one or more of the following properties: - No protection against man-in-the-middle attacks - Same key used for authentication and encryption - Weak message authentication control - No protection against TCP connection closing - Use of weak cipher suites The presence of these properties might allow an attacker to intercept, modify, or tamper with sensitive data. **Example 1:** This Node.js snippet tries to create a server with a secure connection:

```
...
var options = {
port: 443,
path: '/',
key : fs.readFileSync('my-server-key.pem'),
cert : fs.readFileSync('server-cert.pem'),
...
}
https.createServer(options);
...
```

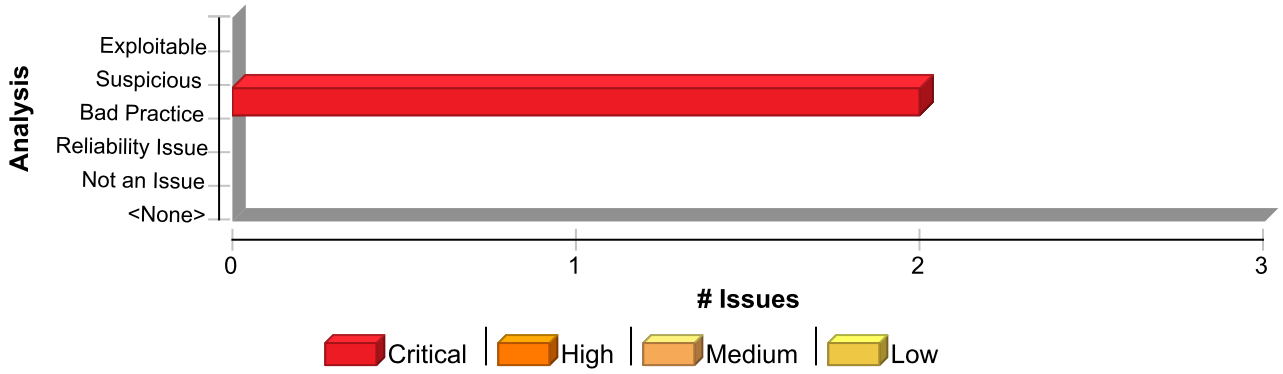
Since Node.js sets the default value of `secureProtocol` to `SSLv23_method`, the server is inherently insecure when `secureProtocol` is not specifically overridden.

Recommendation

Fortify highly recommends forcing the client to use only the most secure protocols. **Example 2:** This Node.js snippet is the same as `Example 1`, except it forces communication over the TLSv1.2 protocol:

```
...
var options = {
port: 443,
path: '/',
secureProtocol: 'TLSv1_2_method',
key : fs.readFileSync('my-server-key.pem'),
cert : fs.readFileSync('server-cert.pem'),
...
}
https.createServer(options);
...
```

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Transport: Weak SSL Protocol	2	0	0	2
Total	**2**	**0**	**0**	**2**

Insecure Transport: Weak SSL Protocol — Critical

Package: Downloads.CodecNetv3.Backend

Downloads/CodecNetv3/Backend/server.js, line 154 (Insecure Transport: Weak SSL Protocol)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FunctionPointerCall: createServer
Enclosing Method: startHTTPSServer()
File: Downloads/CodecNetv3/Backend/server.js:154
Taint Flags:

```
151
152  await database.connect();
153
154  const httpsServer = https.createServer(httpsOptions, app);
155
156  httpsServer.listen(PORT, '0.0.0.0', () => {
157  console.log(`^?^? HTTPS Server is running securely at https://0.0.0.0:${PORT}`);
```

Downloads/CodecNetv3/Backend/index.js, line 74 (Insecure Transport: Weak SSL Protocol)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Insecure Transport: Weak SSL Protocol — Critical

Package: Downloads.CodecNetv3.Backend

Downloads/CodecNetv3/Backend/index.js, line 74 (Insecure Transport: Weak SSL Protocol)

Sink Details

Sink: FunctionPointerCall: createServer
Enclosing Method: startServer()
File: Downloads/CodecNetv3/Backend/index.js:74
Taint Flags:

```
71  ...(sslCaPath && fs.existsSync(sslCaPath) && { ca: fs.readFileSync(sslCaPath) })
72  };
73
74  const httpsServer = https.createServer(httpsOptions, app);
75
76  httpsServer.listen(PORT, () => {
77  console.log(`^?^? HTTPS Server is running securely at https://localhost:${PORT}`);
```

Password Management: Empty Password (6 issues)

Abstract

Empty passwords may compromise system security in a way that is not easy to remedy.

Explanation

It is never a good idea to have an empty password. It also makes fixing the problem extremely difficult once the code is in production. The password cannot be changed without patching the software. If the account protected by the empty password is compromised, the owners of the system must choose between security and availability. **Example 1:** The following code has an empty password to connect to an application and retrieve address book entries:

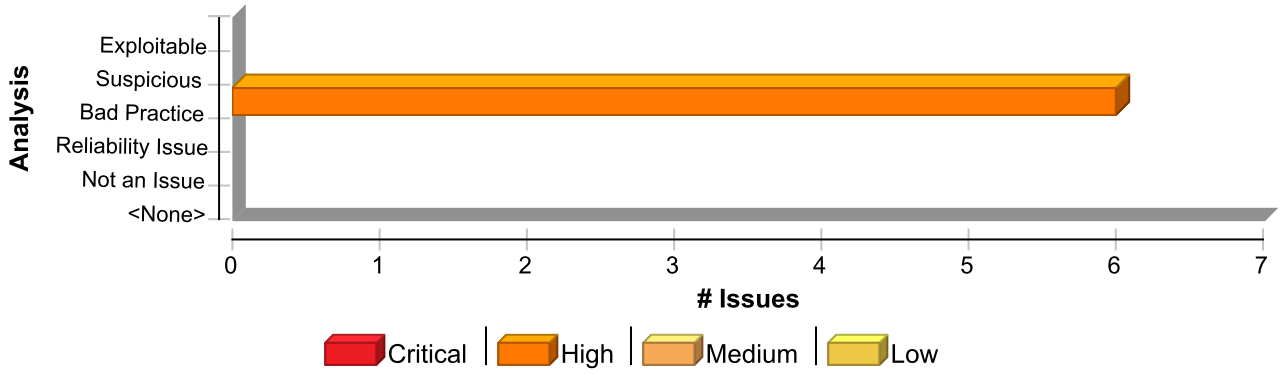
```
...
obj = new XMLHttpRequest();
obj.open('GET','/fetchusers.jsp?id='+form.id.value,'true','scott','');
...
```

This code will run successfully, but anyone can access when they know the username.

Recommendation

Passwords should never be empty and should generally be obfuscated and managed in an external source. Storing passwords in plain text anywhere on the web site allows anyone with sufficient permissions to read and potentially misuse the password. For JavaScript calls that require passwords, it is better to prompt the user for the password at connection time.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Password Management: Empty Password	6	0	0	6
Total	**6**	**0**	**0**	**6**

Password Management: Empty Password — **High**

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 260 (Password Management: Empty Password)

Issue Details

Password Management: Empty Password — High

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 260 (Password Management: Empty Password)

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: UserModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:260
Taint Flags:

```
257  full_name: '',
258  email: '',
259  phone_number: '',
260  password: '',
261  confirmPassword: '',
262  role_id: '',
263  organisation_id: '',
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 336 (Password Management: Empty Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: useEffect0()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:336
Taint Flags:

```
333  full_name: user.full_name || '',
334  email: user.email || '',
335  phone_number: '',
336  password: '',
337  confirmPassword: '',
338  role_id: user.role_id || '',
339  organisation_id: user.organisation_id || '',
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 352 (Password Management: Empty Password)

Issue Details

Password Management: Empty Password — High

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 352 (Password Management: Empty Password)

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: useEffect0()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:352
Taint Flags:

```
349  full_name: '',
350  email: '',
351  phone_number: '',
352  password: '',
353  confirmPassword: '',
354  role_id: '',
355  organisation_id: '',
```

Package: .src.app.(client).user.add

Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx, line 36 (Password Management: Empty Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: ~file_function()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx:36
Taint Flags:

```
33  role: '',
34  level: '',
35  isActive: '',
36  password: '',
37 };
38
39 const FloatInput = ({
```

Password Management: Empty Password — High

Package: .src.app.(client).user.list

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 100 (Password Management: Empty Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: handleEditClick()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:100
Taint Flags:

```
97  role: user.role_id, // Needs to submit role_id!
98  level: user.level || '',
99  is_active: user.is_active,
100  password: '', // Ask user to enter new password, or keep blank
101  });
102  setShowModal(true);
103  };
```

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 45 (Password Management: Empty Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: UserList()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:45
Taint Flags:

```
42  role: '',
43  level: '',
44  is_active: false as boolean | string,
45  password: '',
46  });
47
48  // Fetch users (use independently, so can be re-called)
```

Password Management: Hardcoded Password (3 issues)

Abstract

Hardcoded passwords can compromise system security in a way that is difficult to remedy.

Explanation

Never hardcode passwords. Not only does it expose the password to all of the project's developers, it also makes fixing the problem extremely difficult. After the code is in production, a program patch is probably the only way to change the password. If the account protected by the password is compromised, the organization must choose between security and system availability. **Example 1:** The following URL uses a hardcoded password:

```
...
https://user:secretpassword@example.com
...
```

Example 2: The following ODBC connection string uses a hardcoded password:

```
...
server=Server;database=Database;UID=UserName;PWD=Password;Encrypt=yes;
...
```

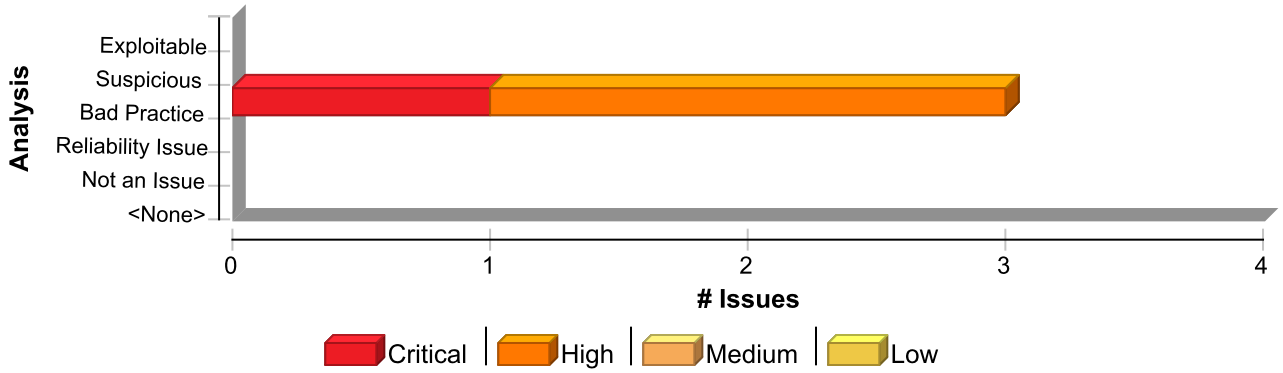
Recommendation

Example 3: The following ODBC connection string addresses the hardcoded password issue in `Example 2` by implementing Integrated Windows Authentication:

```
...
server=Server;database=Database;Trusted_Connection=yes;Encrypt=yes;
...
```

Never use hardcoded passwords. Always obfuscate and manage passwords in an external source. Storing passwords in plain text anywhere on the system enables anyone with sufficient permissions to read and potentially misuse the password.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Password Management: Hardcoded Password	3	0	0	3
Total	**3**	**0**	**0**	**3**

Password Management: Hardcoded Password — Critical

Package: Downloads.CodecNetv3.Backend.routes

Downloads/CodecNetv3/Backend/routes/user.routes.js, line 824 (Password Management: Hardcoded Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Configuration)

Audit Details

Analysis	Bad Practice

Sink Details

Sink:
File: Downloads/CodecNetv3/Backend/routes/user.routes.js:824
Taint Flags:

```
821   * example: 9012534455
822   * password:
823   * type: string
824   * example: $2b$10$3i59ebmuVzq2E7/Wt1oLnOfduKsAAcKhQdmy3cJT131jOodQo8zCC
825   * level:
826   * type: string
827   * example: L1
```

Password Management: Hardcoded Password — High

Package: .src.app.(client).permission.list

Downloads/CodecNetv3/Frontend/src/app/(client)/permission/list/page.tsx, line 37 (Password Management: Hardcoded Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis	Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: fetchUsers()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/permission/list/page.tsx:37
Taint Flags:

```
34  {
35  permission: 'Client',
36  phoneNumber: 9871111222,
37  password: 'client1@123456',
38  role: '68874c0cbb43bw9a1f241',
39  level: 'L1',
40  is_active: true,
```

Password Management: Hardcoded Password — High

Package: .src.app.(client).permission.list

Downloads/CodecNetv3/Frontend/src/app/(client)/permission/list/page.tsx, line 45 (Password Management: Hardcoded Password)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Audit Details

Analysis	Bad Practice

Sink Details

Sink: FieldAccess: password
Enclosing Method: fetchUsers()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/permission/list/page.tsx:45
Taint Flags:

```
42  {
43  permission: 'Manager',
44  phoneNumber: 9873333233,
45  password: 'client2@123456',
46  role: '68874c0cbb43bw9a1f241',
47  level: 'L1',
48  is_active: false,
```

Path Manipulation (12 issues)

Abstract

Allowing user input to control paths used in file system operations could enable an attacker to access or modify otherwise protected system resources.

Explanation

Path manipulation errors occur when the following two conditions are met: 1. An attacker can specify a path used in an operation on the file system. 2. By specifying the resource, the attacker gains a capability that would not otherwise be permitted. For example, the program might give the attacker the ability to overwrite the specified file or run with a configuration controlled by the attacker. **Example 1:** The following code uses input from an HTTP request to create a file name. The programmer has not considered the possibility that an attacker could provide a file name such as "`../../tomcat/conf/server.xml`", which causes the application to delete one of its own configuration files.

```
...
var reportNameParam = "reportName=";
var reportIndex = document.indexOf(reportNameParam);
if (reportIndex < 0) return;
var rName = document.URL.substring(reportIndex+reportNameParam.length);
window.requestFileSystem(window.TEMPORARY, 1024*1024, function(fs) {
fs.root.getFile('/usr/local/apfr/reports/' + rName, {create: false},
function(fileEntry) {
fileEntry.remove(function() {
console.log('File removed.');
}, errorHandler);

}, errorHandler);
}, errorHandler);
```

Example 2: The following code uses input from the local storage to determine which file to open and echo back to the user. If malicious users can change the contents of the local storage, they can use the program to read any file on the system that ends with the extension `.txt`.

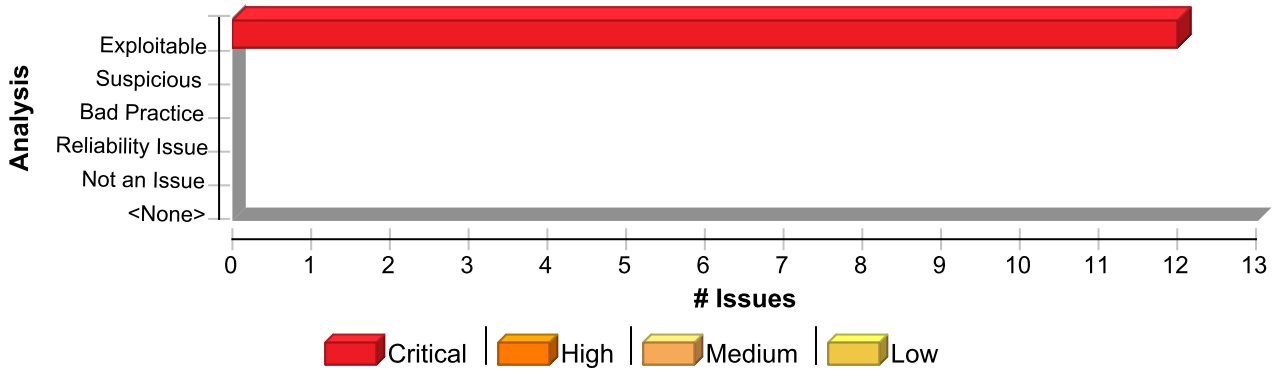
```
...
var filename = localStorage.sub + '.txt';
function oninit(fs) {
fs.root.getFile(filename, {}, function(fileEntry) {
fileEntry.file(function(file) {
var reader = new FileReader();
reader.onloadend = function(e) {
var txtArea = document.createElement('textarea');
txtArea.value = this.result;
document.body.appendChild(txtArea);
};
reader.readAsText(file);
}, errorHandler);
}, errorHandler);
}

window.requestFileSystem(window.TEMPORARY, 1024*1024, oninit, errorHandler);
...
```

Recommendation

The best way to prevent path manipulation is with a level of indirection: create a list of legitimate values

from which the user must select. With this approach, the user-provided input is never used directly to specify the resource name. In some situations this approach is impractical because the set of legitimate resource names is too large or too hard to maintain. Programmers often resort to implementing a deny list in these situations. A deny list is used to selectively reject or escape potentially dangerous characters before using the input. However, any such list of unsafe characters is likely to be incomplete and will almost certainly become out of date. A better approach is to create a list of characters that are permitted to appear in the resource name and accept input composed exclusively of characters in the approved set.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Path Manipulation	12	0	0	12
Total	**12**	**0**	**0**	**12**

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 889 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:841

```
838  * @route GET /api/reports/download/compliance/:filename
839  * @access Token-based (no JWT required, signed token provides authorization)
840  */
841 const downloadComplianceReport = asyncHandler(async (req, res) => {
842  try {
843  const { filename } = req.params;
```

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 889 (Path Manipulation)

```
844  const token = SignedUrlGenerator.extractToken(req);
```

Sink Details

Sink: fs.realpathSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:889
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
886  }
887
888  // Check if file is within reports directory (prevent directory traversal)
889  const realPath = fs.realpathSync(filePath);
890  const realReportsDir = fs.realpathSync(SECURE_REPORTS_DIR);
891  if (!realPath.startsWith(realReportsDir)) {
892  console.error(`^?^? SECURITY: Path traversal attempt blocked: ${filename}`);
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 912 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0.params)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2  Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.createReadStream()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:912
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION,

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 912 (Path Manipulation)

VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
909  res.setHeader('Expires', '0');
910
911  // Stream file to client
912  const fileStream = fs.createReadStream(filePath);
913
914  fileStream.on('error', (error) => {
915  console.error('File stream error:', error);
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 527 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Exploitable

Source Details

Source: lambda(0.body)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2   Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.writeFileSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:527
Taint Flags: CONCATENATED, VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
524
525  // Save PDF to disk
526  const filePath = path.join(storageDir, filename);
527  fs.writeFileSync(filePath, pdfBuffer);
528  console.log(`PDF saved to: ${filePath}`);
529
```

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 527 (Path Manipulation)

```
530  // Get file size
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 521 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:218

```
215 }
216
217 // Generate Report
218 const generateReport = asyncHandler(async (req, res) => {
219  try {
220  const { reportName, frequency = 'weekly', description = '', template =
'executive' } = req.body;
221
```

Sink Details

Sink: fs.mkdirSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:521
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
518  // Create organization directory if it doesn't exist
519  const storageDir = path.join(__dirname, '..', 'storage', 'reports', organizationId);
520  if (!fs.existsSync(storageDir)) {
521  fs.mkdirSync(storageDir, { recursive: true });
522  console.log(`Created directory: ${storageDir}`);
523  }
524
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 912 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 912 (Path Manipulation)

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:841

```
838  * @route GET /api/reports/download/compliance/:filename
839  * @access Token-based (no JWT required, signed token provides authorization)
840  */
841 const downloadComplianceReport = asyncHandler(async (req, res) => {
842  try {
843  const { filename } = req.params;
844  const token = SignedUrlGenerator.extractToken(req);
```

Sink Details

Sink: fs.createReadStream()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:912
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
909  res.setHeader('Expires', '0');
910
911  // Stream file to client
912  const fileStream = fs.createReadStream(filePath);
913
914  fileStream.on('error', (error) => {
915  console.error('File stream error:', error);
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 884 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0)

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 884 (Path Manipulation)

From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:841

```
838   * @route GET /api/reports/download/compliance/:filename
839   * @access Token-based (no JWT required, signed token provides authorization)
840   */
841 const downloadComplianceReport = asyncHandler(async (req, res) => {
842   try {
843   const { filename } = req.params;
844   const token = SignedUrlGenerator.extractToken(req);
```

Sink Details

Sink: fs.existsSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:884
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
881   const filePath = path.join(SECURE_REPORTS_DIR, sanitizedFilename);
882
883   // Check if file exists
884   if (!fs.existsSync(filePath)) {
885   throw new ApiError(404, 'Report not found');
886   }
887
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 884 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Exploitable

Source Details

Source: lambda(0.params)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2   Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
```

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 884 (Path Manipulation)

```
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.existsSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:884
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
881  const filePath = path.join(SECURE_REPORTS_DIR, sanitizedFilename);
882
883  // Check if file exists
884  if (!fs.existsSync(filePath)) {
885  throw new ApiError(404, 'Report not found');
886  }
887
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 520 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:218

```
215 }
216
217 // Generate Report
218 const generateReport = asyncHandler(async (req, res) => {
219  try {
220  const { reportName, frequency = 'weekly', description = '', template = 'executive' } = req.body;
221
```

Sink Details

Sink: fs.existsSync()

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 520 (Path Manipulation)

Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:520
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
517
518  // Create organization directory if it doesn't exist
519  const storageDir = path.join(__dirname, '..', 'storage', 'reports', organizationId);
520  if (!fs.existsSync(storageDir)) {
521  fs.mkdirSync(storageDir, { recursive: true });
522  console.log(`Created directory: ${storageDir}`);
523  }
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 520 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2  Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.existsSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:520
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
517
518  // Create organization directory if it doesn't exist
519  const storageDir = path.join(__dirname, '..', 'storage', 'reports', organizationId);
520  if (!fs.existsSync(storageDir)) {
```

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 520 (Path Manipulation)

```
521  fs.mkdirSync(storageDir, { recursive: true });
522  console.log(`Created directory: ${storageDir}`);
523  }
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 889 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0.params)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2   Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.realpathSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:889
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
886  }
887
888  // Check if file is within reports directory (prevent directory traversal)
889  const realPath = fs.realpathSync(filePath);
890  const realReportsDir = fs.realpathSync(SECURE_REPORTS_DIR);
891  if (!realPath.startsWith(realReportsDir)) {
892  console.error(`^?^? SECURITY: Path traversal attempt blocked: ${filename}`);
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 527 (Path Manipulation)

Issue Details

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 527 (Path Manipulation)

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:218

```
215 }
216
217 // Generate Report
218 const generateReport = asyncHandler(async (req, res) => {
219  try {
220  const { reportName, frequency = 'weekly', description = '', template = 'executive' } = req.body;
221
```

Sink Details

Sink: fs.writeFileSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:527
Taint Flags: CONCATENATED, VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
524
525  // Save PDF to disk
526  const filePath = path.join(storageDir, filename);
527  fs.writeFileSync(filePath, pdfBuffer);
528  console.log(`PDF saved to: ${filePath}`);
529
530  // Get file size
```

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 521 (Path Manipulation)

Issue Details

Kingdom: Input Validation and Representation
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Source Details

Path Manipulation — Critical

Package: Downloads.CodecNetv3.Backend.controllers

Downloads/CodecNetv3/Backend/controllers/reports.controller.js, line 521 (Path Manipulation)

Source: lambda(0)
From: lambda
File: Downloads/CodecNetv3/Backend/utils/asyncHandler.js:1

```
1 export const asyncHandler = (fn) => (req, res, next) => {
2  Promise.resolve(fn(req, res, next)).catch(next);
3 };
4
5 undefined
6 undefined
7 undefined
```

Sink Details

Sink: fs.mkdirSync()
Enclosing Method: lambda()
File: Downloads/CodecNetv3/Backend/controllers/reports.controller.js:521
Taint Flags: VALIDATED_CLIENT_SIDE_TEMPLATE_INJECTION, VALIDATED_CROSS_SITE_SCRIPTING_DOM, VALIDATED_CROSS_SITE_SCRIPTING_SELF, WEB, XSS

```
518  // Create organization directory if it doesn't exist
519  const storageDir = path.join(__dirname, '..', 'storage', 'reports', organizationId);
520  if (!fs.existsSync(storageDir)) {
521  fs.mkdirSync(storageDir, { recursive: true });
522  console.log(`Created directory: ${storageDir}`);
523  }
524
```

Privacy Violation (23 issues)

Abstract

Mishandling private information, such as customer passwords or social security numbers, can compromise user privacy and is often illegal.

Explanation

Privacy violations occur when: 1. Private user information enters the program. 2. The data is written to an external location, such as the console, file system, or network. **Example 1:** The following code stores user's plain text password to the local storage.

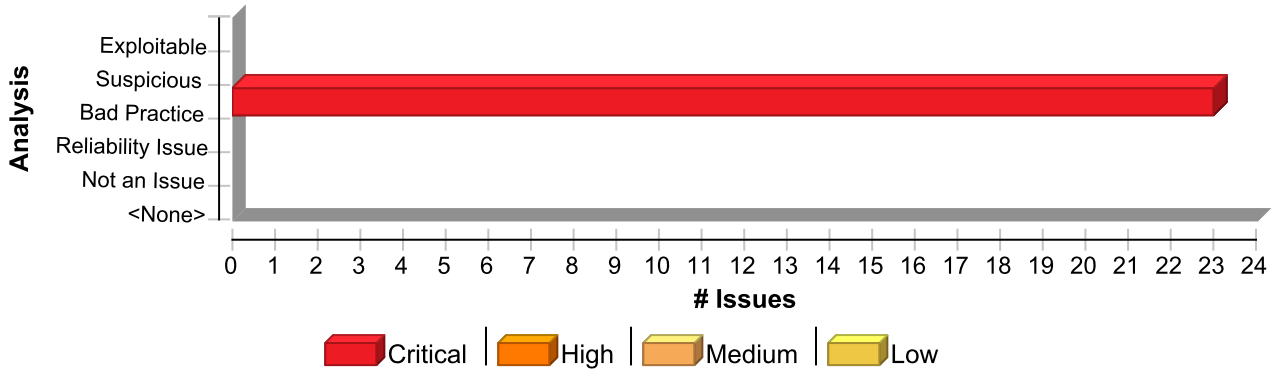
```
localStorage.setItem('password', password);
```

Although many developers treat the local storage as a safe location for data, it should not be trusted implicitly, particularly when privacy is a concern. Private data can enter a program in a variety of ways: - Directly from the user in the form of a password or personal information - Accessed from a database or other data store by the application - Indirectly from a partner or other third party Sometimes data that is not labeled as private can have a privacy implication in a different context. For example, student identification numbers are usually not considered private because there is no explicit and publicly-available mapping to an individual student's personal information. However, if a school generates identification numbers based on student social security numbers, then the identification numbers should be considered private. Security and privacy concerns often seem to compete with each other. From a security perspective, you should record all important operations so that any anomalous activity can later be identified. However, when private data is involved, this practice can create risk. Although there are many ways in which private data can be handled unsafely, a common risk stems from misplaced trust. Programmers often trust the operating environment in which a program runs, and therefore believe that it is acceptable to store private information on the file system, in the registry, or in other locally-controlled resources. However, even if access to certain resources is restricted, this does not guarantee that the individuals who do have access can be trusted. For example, in 2004, an unscrupulous employee at AOL sold approximately 92 million private customer email addresses to a spammer marketing an offshore gambling web site [1]. In response to such high-profile exploits, the collection and management of private data is becoming increasingly regulated. Depending on its location, the type of business it conducts, and the nature of any private data it handles, an organization may be required to comply with one or more of the following federal and state regulations: - Safe Harbor Privacy Framework [3] - Gramm-Leach Bliley Act (GLBA) [4] - Health Insurance Portability and Accountability Act (HIPAA) [5] - California SB-1386 [6] Despite these regulations, privacy violations continue to occur with alarming frequency.

Recommendation

When security and privacy demands clash, privacy should usually be given the higher priority. To accomplish this and still maintain required security information, cleanse any private information before it exits the program. To enforce good privacy management, develop and strictly adhere to internal privacy guidelines. The guidelines should specifically describe how an application should handle private data. If your organization is regulated by federal or state law, ensure that your privacy guidelines are sufficiently strenuous to meet the legal requirements. Even if your organization is not regulated, you must protect private information or risk losing customer confidence. The best policy with respect to private data is to minimize its exposure. Applications, processes, and employees should not be granted access to any private data unless the access is required for the tasks that they are to perform. Just as the principle of least privilege dictates that no operation should be performed with more than the necessary privileges, access to private data should be restricted to the smallest possible group.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Privacy Violation	23	0	0	23
Total	**23**	**0**	**0**	**23**

Privacy Violation — Critical

Package: .src.app.(client).agents

Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx, line 1262 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read adminPassword
From: AgentsPage
File: Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx:1262

```
1259  <label className="text-sm font-medium text-gray-700 dark:text-
gray-300">Super Admin Password</label>
1260  <input
1261  type="password"
1262  value={adminPassword}
1263  onChange={(e) => setAdminPassword(e.target.value)}
1264  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600
rounded-md focus:ring-blue-500 focus:border-blue-500 dark:bg-gray-700
dark:text-white"
1265  placeholder="Enter super admin password"
```

Sink Details

Sink: Assignment to value
Enclosing Method: AgentsPage()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx:1262
Taint Flags: PRIVATE

Privacy Violation — Critical

Package: .src.app.(client).agents

Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx, line 1262 (Privacy Violation)

```
1259  <label className="text-sm font-medium text-gray-700 dark:text-gray-300">Super Admin Password</label>
1260  <input
1261  type="password"
1262  value={adminPassword}
1263  onChange={(e) => setAdminPassword(e.target.value)}
1264  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md focus:ring-blue-500 focus:border-blue-500 dark:bg-gray-700 dark:text-white"
1265  placeholder="Enter super admin password"
```

Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx, line 1370 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read adminPassword
From: AgentsPage
File: Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx:1370

```
1367  <label className="text-sm font-medium text-gray-700 dark:text-gray-300">Super Admin Password</label>
1368  <input
1369  type="password"
1370  value={adminPassword}
1371  onChange={(e) => setAdminPassword(e.target.value)}
1372  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md focus:ring-blue-500 focus:border-blue-500 dark:bg-gray-700 dark:text-white"
1373  placeholder="Enter super admin password"
```

Sink Details

Sink: Assignment to value
Enclosing Method: AgentsPage()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx:1370
Taint Flags: PRIVATE

```
1367  <label className="text-sm font-medium text-gray-700 dark:text-gray-300">Super Admin Password</label>
1368  <input
1369  type="password"
1370  value={adminPassword}
```

Privacy Violation — Critical

Package: .src.app.(client).agents

Downloads/CodecNetv3/Frontend/src/app/(client)/agents/page.tsx, line 1370 (Privacy Violation)

```
1371  onChange={(e) => setAdminPassword(e.target.value)}
1372  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md focus:ring-blue-500 focus:border-blue-500 dark:bg-gray-700 dark:text-white"
1373  placeholder="Enter super admin password"
```

Package: .src.app.(client).overview

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 1453 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read formData.wazuh_dashboard_password
From: AddClientModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1453

```
1450  <input
1451  type="password"
1452  required
1453  value={formData.wazuh_dashboard_password}
1454  onChange={(e) => {
1455  clearErrors();
1456  setFormData({ ...formData, wazuh_dashboard_password: e.target.value });
```

Sink Details

Sink: Assignment to value
Enclosing Method: AddClientModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1453
Taint Flags: PRIVATE

```
1450  <input
1451  type="password"
1452  required
1453  value={formData.wazuh_dashboard_password}
1454  onChange={(e) => {
1455  clearErrors();
1456  setFormData({ ...formData, wazuh_dashboard_password: e.target.value });
```

Privacy Violation — Critical

Package: .src.app.(client).overview

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 1338 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read formData.wazuh_indexer_password
From: AddClientModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1338

```
1335  <input
1336  type="password"
1337  required
1338  value={formData.wazuh_indexer_password}
1339  onChange={(e) => {
1340  clearErrors();
1341  setFormData({ ...formData, wazuh_indexer_password: e.target.value });
```

Sink Details

Sink: Assignment to value
Enclosing Method: AddClientModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1338
Taint Flags: PRIVATE

```
1335  <input
1336  type="password"
1337  required
1338  value={formData.wazuh_indexer_password}
1339  onChange={(e) => {
1340  clearErrors();
1341  setFormData({ ...formData, wazuh_indexer_password: e.target.value });
```

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 1223 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read formData.wazuh_manager_password

Privacy Violation — Critical

Package: .src.app.(client).overview

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 1223 (Privacy Violation)

From: AddClientModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1223

```
1220  <input
1221  type="password"
1222  required
1223  value={formData.wazuh_manager_password}
1224  onChange={(e) => {
1225  clearErrors();
1226  setFormData({ ...formData, wazuh_manager_password: e.target.value });
```

Sink Details

Sink: Assignment to value
Enclosing Method: AddClientModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:1223
Taint Flags: PRIVATE

```
1220  <input
1221  type="password"
1222  required
1223  value={formData.wazuh_manager_password}
1224  onChange={(e) => {
1225  clearErrors();
1226  setFormData({ ...formData, wazuh_manager_password: e.target.value });
```

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 354 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read superAdminPassword
From: DeleteConfirmationModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:354

```
351  <input
352  type="password"
353  required
354  value={superAdminPassword}
355  onChange={(e) => {
```

Privacy Violation — Critical

Package: .src.app.(client).overview

Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx, line 354 (Privacy Violation)

```
356   setSuperAdminPassword(e.target.value);
357   setError('');
```

Sink Details

Sink: Assignment to value
Enclosing Method: DeleteConfirmationModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/overview/page.tsx:354
Taint Flags: PRIVATE

```
351   <input
352   type="password"
353   required
354   value={superAdminPassword}
355   onChange={(e) => {
356   setSuperAdminPassword(e.target.value);
357   setError('');
```

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 757 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read form.confirmPassword
From: UserModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:757

```
754   <input
755   type="password"
756   maxLength={128}
757   value={form.confirmPassword}
758   onChange={(e) => handleInputChange('confirmPassword', e.target.value)}
759   className={`w-full p-3 border-2 rounded-xl bg-white dark:bg-gray-800 text-gray-900 dark:text-white placeholder-gray-500 dark:placeholder-gray-400 focus:ring-2 transition-all duration-200 ${
760   validationErrors.confirmPassword
```

Sink Details

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 757 (Privacy Violation)

Sink: Assignment to value
Enclosing Method: UserModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:757
Taint Flags: PRIVATE

```
754  <input
755  type="password"
756  maxLength={128}
757  value={form.confirmPassword}
758  onChange={(e) => handleInputChange('confirmPassword', e.target.value)}
759  className={`w-full p-3 border-2 rounded-xl bg-white dark:bg-gray-800 text-gray-900
dark:text-white placeholder-gray-500 dark:placeholder-gray-400 focus:ring-2 transition-all
duration-200 ${
760  validationErrors.confirmPassword
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 527 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read payload.confirmPassword
From: handleSubmit
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:491

```
488  }
489
490  const payload: any = { ...form };
491  delete payload.confirmPassword;
492
493  // Clean phone number
494  if (payload.phone_number) {
```

Sink Details

Sink: ~JS_Generic.log()
Enclosing Method: handleSubmit()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:527
Taint Flags: PRIVATE

```
524  const apiUrl = isEditing ? `http://localhost:5000/api/users/${user?.id}` : 'http://
localhost:5000/api/users';
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 527 (Privacy Violation)

```
525  const method = isEditing ? 'PUT' : 'POST';
526
527  console.log('Making request to:', apiUrl, 'with payload:', payload);
528
529  const response = await fetch(apiUrl, {
530  method: method,
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 316 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read passwords.new
From: ChangePasswordForm
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:316

```
313  <input
314  type="password"
315  required
316  value={passwords.new}
317  onChange={(e) => setPasswords({ ...passwords, new: e.target.value })}
318  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500 focus:border-transparent"
319  />
```

Sink Details

Sink: Assignment to value
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:316
Taint Flags: PRIVATE

```
313  <input
314  type="password"
315  required
316  value={passwords.new}
317  onChange={(e) => setPasswords({ ...passwords, new: e.target.value })}
318  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 316 (Privacy Violation)

```
focus:border-transparent"
319  />
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 329 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read passwords.confirm
From: ChangePasswordForm
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:329

```
326  <input
327  type="password"
328  required
329  value={passwords.confirm}
330  onChange={(e) => setPasswords({ ...passwords, confirm: e.target.value })}
331  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500 focus:border-transparent"
332  />
```

Sink Details

Sink: Assignment to value
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:329
Taint Flags: PRIVATE

```
326  <input
327  type="password"
328  required
329  value={passwords.confirm}
330  onChange={(e) => setPasswords({ ...passwords, confirm: e.target.value })}
331  className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500 focus:border-transparent"
332  />
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 731 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read form.password
From: UserModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:731

```
728  <input
729  type="password"
730  maxLength={128}
731  value={form.password}
732  onChange={(e) => handleInputChange('password', e.target.value)}
733  className={`w-full p-3 border-2 rounded-xl bg-white dark:bg-gray-800 text-gray-900 dark:text-white placeholder-gray-500 dark:placeholder-gray-400 focus:ring-2 transition-all duration-200 ${
734  validationErrors.password
```

Sink Details

Sink: Assignment to value
Enclosing Method: UserModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:731
Taint Flags: PRIVATE

```
728  <input
729  type="password"
730  maxLength={128}
731  value={form.password}
732  onChange={(e) => handleInputChange('password', e.target.value)}
733  className={`w-full p-3 border-2 rounded-xl bg-white dark:bg-gray-800 text-gray-900 dark:text-white placeholder-gray-500 dark:placeholder-gray-400 focus:ring-2 transition-all duration-200 ${
734  validationErrors.password
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 751 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 751 (Privacy Violation)

Audit Details

Analysis Bad Practice

Source Details

Source: Read form.password
From: UserModal
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:752

```
749
750   <div>
751   <label className="block text-sm font-medium text-gray-700 dark:text-gray-300 mb-2">
752   Confirm Password {(!isEditing || form.password) && '*'}
753   </label>
754   <input
755   type="password"
```

Sink Details

Sink: Assignment to textContent
Enclosing Method: UserModal()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:751
Taint Flags: PRIVATE

```
748   </div>
749
750   <div>
751   <label className="block text-sm font-medium text-gray-700 dark:text-gray-300 mb-2">
752   Confirm Password {(!isEditing || form.password) && '*'}
753   </label>
754   <input
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 303 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read passwords.current
From: ChangePasswordForm
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySe

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 303 (Privacy Violation)

ttings.tsx:303

```
300   <input
301   type="password"
302   required
303   value={passwords.current}
304   onChange={(e) => setPasswords({ ...passwords, current: e.target.value })}
305   className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500 focus:border-transparent"
306   />
```

Sink Details

Sink: Assignment to value
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:303
Taint Flags: PRIVATE

```
300   <input
301   type="password"
302   required
303   value={passwords.current}
304   onChange={(e) => setPasswords({ ...passwords, current: e.target.value })}
305   className="w-full px-3 py-2 border border-gray-300 dark:border-gray-600 rounded-md bg-white dark:bg-gray-700 text-gray-900 dark:text-white focus:ring-2 focus:ring-blue-500 focus:border-transparent"
306   />
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 527 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read payload.password
From: handleSubmit
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:520

```
517   }
518
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx, line 527 (Privacy Violation)

```
519  if (isEditing && !form.password) {
520  delete payload.password;
521  }
522
523  try {
```

Sink Details

Sink: ~JS_Generic.log()
Enclosing Method: handleSubmit()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/UserManagement.tsx:527
Taint Flags: PRIVATE

```
524  const apiUrl = isEditing ? `http://localhost:5000/api/users/${user?.id}` : 'http://
localhost:5000/api/users';
525  const method = isEditing ? 'PUT' : 'POST';
526
527  console.log('Making request to:', apiUrl, 'with payload:', payload);
528
529  const response = await fetch(apiUrl, {
530  method: method,
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 291 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read ChangePasswordForm
From: SecuritySettings
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:140

```
137  </div>
138
139  {showChangePassword && (
140  <ChangePasswordForm onClose={() => setShowChangePassword(false)} />
141  )}
142  </div>
143  </div>
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 291 (Privacy Violation)

Sink Details

Sink: Assignment to textContent
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:291
Taint Flags: PRIVATE

```
288
289  return (
290  <div className="border border-gray-200 dark:border-gray-700 rounded-lg p-4 bg-gray-50 dark:bg-gray-900/50">
291  <h5 className="text-sm font-medium text-gray-900 dark:text-white mb-4">
292  Change Password
293  </h5>
294
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 297 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read ChangePasswordForm
From: SecuritySettings
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:140

```
137  </div>
138
139  {showChangePassword && (
140  <ChangePasswordForm onClose={() => setShowChangePassword(false)} />
141  )}
142  </div>
143  </div>
```

Sink Details

Sink: Assignment to textContent
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:297
Taint Flags: PRIVATE

```
294
295  <form onSubmit={handleSubmit} className="space-y-4">
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 297 (Privacy Violation)

```
296  <div>
297  <label className="block text-sm font-medium text-gray-700 dark:text-gray-300 mb-1">
298  Current Password
299  </label>
300  <input
```

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 310 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read ChangePasswordForm
From: SecuritySettings
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:140

```
137  </div>
138
139  {showChangePassword && (
140  <ChangePasswordForm onClose={() => setShowChangePassword(false)} />
141  )}
142  </div>
143  </div>
```

Sink Details

Sink: Assignment to textContent
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:310
Taint Flags: PRIVATE

```
307  </div>
308
309  <div>
310  <label className="block text-sm font-medium text-gray-700 dark:text-gray-300 mb-1">
311  New Password
312  </label>
313  <input
```

Privacy Violation — Critical

Package: .src.app.(client).settings.components

Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx, line 323 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read ChangePasswordForm
From: SecuritySettings
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:140

```
137   </div>
138
139   {showChangePassword && (
140   <ChangePasswordForm onClose={() => setShowChangePassword(false)} />
141   )}
142   </div>
143   </div>
```

Sink Details

Sink: Assignment to textContent
Enclosing Method: ChangePasswordForm()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/settings/components/SecuritySettings.tsx:323
Taint Flags: PRIVATE

```
320   </div>
321
322   <div>
323   <label className="block text-sm font-medium text-gray-700 dark:text-gray-300 mb-1">
324   Confirm New Password
325   </label>
326   <input
```

Package: .src.app.(client).user.add

Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx, line 60 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Privacy Violation — Critical

Package: .src.app.(client).user.add

Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx, line 60 (Privacy Violation)

Source Details

Source: Read formData.password
From: AddUserPage
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx:225

```
222   type="password"
223   name="password"
224   placeholder="Password"
225   value={formData.password}
226   onChange={handleChange}
227   />
228
```

Sink Details

Sink: Assignment to value
Enclosing Method: FloatInput()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/add/page.tsx:60
Taint Flags: PRIVATE

```
57  <input
58  type={type}
59  name={name}
60  value={value}
61  onChange={onChange}
62  pattern={pattern}
63  title={title}
```

Package: .src.app.(client).user.list

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 330 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Source Details

Source: Read formFields.password
From: UserList
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:330

```
327   type="password"
328   className="w-full px-4 py-2 rounded-md bg-gray-800 text-white"
```

Privacy Violation — Critical

Package: .src.app.(client).user.list

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 330 (Privacy Violation)

```
329   name="password"
330   value={formFields.password}
331   onChange={handleFormChange}
332   placeholder="Password (required)"
333   required
```

Sink Details

Sink: Assignment to value
Enclosing Method: UserList()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:330
Taint Flags: PRIVATE

```
327   type="password"
328   className="w-full px-4 py-2 rounded-md bg-gray-800 text-white"
329   name="password"
330   value={formFields.password}
331   onChange={handleFormChange}
332   placeholder="Password (required)"
333   required
```

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 335 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read formFields.password
From: UserList
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:330

```
327   type="password"
328   className="w-full px-4 py-2 rounded-md bg-gray-800 text-white"
329   name="password"
330   value={formFields.password}
331   onChange={handleFormChange}
332   placeholder="Password (required)"
333   required
```

Sink Details

Privacy Violation — Critical

Package: .src.app.(client).user.list

Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx, line 335 (Privacy Violation)

Sink: Assignment to textContent
Enclosing Method: UserList()
File: Downloads/CodecNetv3/Frontend/src/app/(client)/user/list/page.tsx:335
Taint Flags: PRIVATE

```
332  placeholder="Password (required)"
333  required
334  />
335  <label className="text-white flex items-center gap-2">
336  Is Active
337  <input
338  type="checkbox"
```

Package: .src.app.login

Downloads/CodecNetv3/Frontend/src/app/login/page.tsx, line 171 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read password
From: LoginPage
File: Downloads/CodecNetv3/Frontend/src/app/login/page.tsx:171

```
168  </div>
169  <input
170  type={showPassword ? 'text' : 'password'}
171  value={password}
172  onChange={(e) => setPassword(e.target.value)}
173  autoComplete="current-password"
174  className="w-full pl-12 pr-14 py-4 bg-gray-700/50 backdrop-blur-sm
border border-gray-600/50 rounded-xl text-white placeholder-gray-500
focus:outline-none focus:ring-2 focus:ring-blue-500/50 focus:border-
blue-500/50 transition-all duration-300"
```

Sink Details

Sink: Assignment to value
Enclosing Method: LoginPage()
File: Downloads/CodecNetv3/Frontend/src/app/login/page.tsx:171
Taint Flags: PRIVATE

Privacy Violation — Critical

Package: .src.app.login

Downloads/CodecNetv3/Frontend/src/app/login/page.tsx, line 171 (Privacy Violation)

```
168   </div>
169   <input
170   type={showPassword ? 'text' : 'password'}
171   value={password}
172   onChange={(e) => setPassword(e.target.value)}
173   autoComplete="current-password"
174   className="w-full pl-12 pr-14 py-4 bg-gray-700/50 backdrop-blur-sm border border-
gray-600/50 rounded-xl text-white placeholder-gray-500 focus:outline-none focus:ring-2
focus:ring-blue-500/50 focus:border-blue-500/50 transition-all duration-300"
```

Package: Downloads.CodecNetv3.Backend.scripts

Downloads/CodecNetv3/Backend/scripts/test-wazuh-auth.js, line 23 (Privacy Violation)

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Data Flow)

Audit Details

Analysis	Bad Practice

Source Details

Source: Read codecOrg.wazuh_manager_password
From: testWazuhAuth
File: Downloads/CodecNetv3/Backend/scripts/test-wazuh-auth.js:23

```
20  console.log('Organisation:', codecOrg.organisation_name);
21  console.log('Wazuh IP:', codecOrg.wazuh_manager_ip);
22  console.log('Username:', codecOrg.wazuh_manager_username);
23  console.log('Password:', codecOrg.wazuh_manager_password);
24
25  const host = `https://${codecOrg.wazuh_manager_ip}:$
{codecOrg.wazuh_manager_port || 55000}`;
26  const auth = Buffer.from(`${codecOrg.wazuh_manager_username}:$
{codecOrg.wazuh_manager_password}`).toString('base64');
```

Sink Details

Sink: ~JS_Generic.log()
Enclosing Method: testWazuhAuth()
File: Downloads/CodecNetv3/Backend/scripts/test-wazuh-auth.js:23
Taint Flags: PRIVATE

```
20  console.log('Organisation:', codecOrg.organisation_name);
21  console.log('Wazuh IP:', codecOrg.wazuh_manager_ip);
22  console.log('Username:', codecOrg.wazuh_manager_username);
23  console.log('Password:', codecOrg.wazuh_manager_password);
```

Privacy Violation — Critical

Package: Downloads.CodecNetv3.Backend.scripts

Downloads/CodecNetv3/Backend/scripts/test-wazuh-auth.js, line 23 (Privacy Violation)

```
24
25  const host = `https://${codecOrg.wazuh_manager_ip}:${codecOrg.wazuh_manager_port || 55000}
`;
26  const auth = Buffer.from(`${codecOrg.wazuh_manager_username}:$
{codecOrg.wazuh_manager_password}`).toString('base64');
```